BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

CLAIMS

1. A method for determining whether computer code contains malicious code, said method comprising the steps of:

optimizing the computer code to produce optimized code; and

subjecting the optimized code to a malicious code detection protocol.

2. The method of claim 1 wherein the malicious code detection protocol is a protocol from the group of protocols consisting of pattern matching, emulation, checksumming, heuristics, tracing, X-raying, and algorithmic scanning.

3. The method of claim 1 wherein the optimizing step comprises performing at least one technique from the group of techniques consisting of constant folding, copy propagation, non-obvious dead code elimination, code motion, peephole optimization, abstract interpretation, instruction specialization, and control flow graph reduction.

4. The method of claim 3 wherein at least two of said techniques are combined synergistically.

5. The method of claim 1 wherein the computer code is polymorphic code comprising a decryption loop and a body; and

the optimizing step comprises optimizing just the decryption loop.

6. A method for determining whether computer code having a decryption loop and a body contains malicious code, said method comprising the steps of:

optimizing the decryption loop to produce optimized loop code;

performing a malicious code detection procedure on the optimized loop code;

optimizing the body to produce optimized body code; and

subjecting the optimized body code to a malicious code detection protocol.

7. The method of claim 6 wherein the malicious code detection procedure is a procedure from the group of procedures consisting of pattern matching, emulation, checksumming, heuristics, tracing, and algorithmic scanning.

8. The method of claim 6 wherein the malicious code detection protocol is a protocol from the group of protocols consisting of pattern matching, emulation, checksumming, heuristics, tracing, X-raying, and algorithmic scanning.

9. The method of claim 6 wherein the step of optimizing the body comprises using at least one output from the group of steps consisting of optimizing the decryption loop and performing a malicious code detection procedure on the optimized loop code.

10. The method of claim 6 wherein, when the step of performing a malicious code detection procedure on the optimized loop code indicates the presence of malicious code in the computer code, the steps of optimizing the body and subjecting the optimized body code to a malicious code detection protocol are aborted.

11. The method of claim 6 further comprising the additional step of, after the step of performing a malicious code detection procedure on the optimized loop code, revealing an encrypted body.

12. The method of claim 11 wherein the step of revealing an encrypted body comprises emulating the optimized loop code.

13. The method of claim 11 wherein the step of revealing an encrypted body comprises applying a key gleaned from the optimized loop code.

14. A method for optimizing computer code that is suspected of containing malicious code, said method comprising the steps of:

performing a forward pass operation;

performing a backward pass operation;

performing a control flow graph reduction; and

iterating the above three steps a plurality of times.

15. The method of claim 14 wherein the iteration of the three steps stops after either:

a preselected number of iterations; or

observing that no optimizations of the computer code were performed in the most recent iteration.

16. The method of claim 14 further comprising the step of performing a code motion procedure, wherein the four steps are iterated a plurality of times.

17. The method of claim 14 wherein the forward pass operation comprises at least one of the following steps:

peephole optimization;

constant folding;

copy propagation;

forward computations related to abstract interpretation; and

instruction specialization.

18. The method of claim 14 wherein the backward pass operation comprises at least one of the steps of backward computations related to abstract interpretation and local dead code elimination.

19. The method of claim 18 wherein the backward pass operation comprises the additional step of global dead code elimination.

20. Apparatus for countering malicious computer code, said apparatus comprising:

a peephole optimizer;

coupled to the peephole optimizer, a state tracking module; and

coupled to the peephole optimizer and to the state tracking module, an instruction specialization module.

21. The apparatus of claim 20 further comprising a virtual state memory module coupled to the state tracking module.

22. The apparatus of claim 20 further comprising a driver module coupled to the instruction specialization module and to the state tracking module.

23. The apparatus of claim 20 wherein the peephole optimizer comprises an instruction reordering module.

24. A computer-readable medium containing computer program instructions for determining whether computer code contains malicious code, said computer program instructions performing the steps of:

optimizing the computer code to produce optimized code; and

subjecting the optimized code to a malicious code detection protocol.

25. The computer-readable medium of claim 24 wherein the malicious code detection protocol is a protocol from the group of protocols consisting of pattern matching, emulation, checksumming, heuristics, tracing, X-raying, and algorithmic scanning.

26. The computer-readable medium of claim 24 wherein the optimizing step comprises performing at least one technique from the group of techniques consisting of constant folding, copy propagation, non-obvious dead code elimination, code motion, peephole optimization, abstract interpretation, instruction specialization, and control flow graph reduction.

27. A method for determining whether computer code contains malicious code, said method comprising the steps of:

performing a dead code elimination procedure on the computer code;

noting the amount of dead code eliminated during the dead code elimination procedure; and

when the amount of dead code eliminated during the dead code elimination procedure exceeds a preselected dead code threshold, declaring a suspicion of malicious code in the computer code.